Annales Mathematicae et Informaticae
61 (2025) pp. 43–54
DOI: 10.33039/ami.2025.10.005
URL: https://ami.uni-eszterhazy.hu

Artificial Intelligence for interpreting static human arm signals

Milán Zsolt Bagladi[a*]

[a]Faculty of Informatics, Eötvös Loránd University
hhpw8b@inf.elte.hu

Abstract. This paper presents a method for static arm signal recognition using OpenPose-based keypoint estimation, keypoint normalization, and two distinct classification approaches: K-means clustering and a neural network classifier. The system works with a simple camera setup and generalizes across users. A keypoint normalization technique is used to handle differences in body size and camera distance. To improve robustness against body rotation, we introduce a technique for generating artificially rotated training data using 3D keypoint reconstruction. The recognition models were trained and evaluated on a custom dataset of nine gestures, while rotation robustness was tested on a representative subset of three gestures. Results show that both models maintain high accuracy and efficiency even under moderate rotation.

Keywords: Arm Gesture Recognition, Static Gestures, OpenPose, Keypoint Normalization, K-means Clustering, Neural Networks, Data Augmentation, 3D Reconstruction, Human-Computer Interaction, Rotation Robustness

AMS Subject Classification: Primary: 68T07, Secondary: 68T40

1. Introduction

Human arm gestures are a natural and intuitive means of communication, frequently used in everyday situations ranging from traffic control to human-robot interaction. While easily interpreted by humans, the automatic recognition of such gestures remains a challenging task for computer systems due to the variability in body types, camera perspectives, and environmental conditions.

*Special thanks to my supervisor, Dr. László Gulyás[a] for his support and Gergő Szalay[a] for his guidance.

Accepted: October 8, 2025
Published online: October 28, 2025

A particularly important class of gestures is static arm signals, in which the meaning is conveyed by a single body pose, independent of motion or temporal context. These static signals are prevalent in domains such as traffic management, where police officers use arm positions to direct vehicles, or in aviation, where ground crews communicate using standardized poses. In such safety-critical applications, accurate and real-time recognition is essential.

Recent advances in computer vision, especially in human pose estimation, have made it possible to extract structural information about the human body from images. However, interpreting this data for gesture classification still requires robust and efficient algorithms. Many existing solutions rely on expensive hardware or are sensitive to variations in camera angles and user appearances.

In this paper, we propose a lightweight, camera-based solution for recognizing static human arm gestures. Our approach uses OpenPose for keypoint extraction, followed by normalization to handle variations in camera distance and body proportions. We explore both unsupervised (K-means clustering) and supervised (neural network) classification methods. To improve robustness against changes in camera orientation, we introduce a novel data augmentation technique using artificially rotated skeletons. The methods were evaluated on a custom dataset of nine gestures from multiple individuals, with rotation robustness tested on a subset of three gestures rotated up to 45°. The results demonstrate that both classification models achieve high accuracy and fast inference even under moderate rotation, validating the practicality of our approach.

2. Problem statement

The primary objective of this work is to develop a robust and efficient system for recognizing a predefined set of static human arm signals from a single 2D image. Such a system is essential for applications in traffic control, logistics, and human-robot interaction, where clear and immediate interpretation of human signals is critical.

The core technical challenge is to create a classifier that is invariant to several factors:

- **Viewpoint Variation:** The system must reliably identify gestures even when the person is not directly facing the camera. A key goal is to maintain high accuracy under moderate body rotations (e.g., up to 45°).
- **User-Specific Differences:** The model must generalize across individuals with different body proportions, sizes, and minor variations in gesture execution.
- **Scale and Position:** Recognition should be independent of the person's distance from the camera and their position within the frame.

Furthermore, the solution must be practical, operating in real-time with a standard monocular camera, without specialized hardware. This paper addresses these challenges by leveraging 2D pose keypoints and techniques for robustness and generalization.

3. Related work

The field of gesture recognition has seen significant advancement in recent years, particularly through the integration of pose estimation and machine learning techniques.

An early approach to arm gesture recognition using convolutional neural networks was proposed by Mathe et al. [8], where gestures were classified based on key features extracted from depth and color images. Their work demonstrated the feasibility of using CNNs for classifying human arm gestures with reasonable accuracy, though it primarily focused on dynamic inputs and required more constrained settings.

A major breakthrough in human pose estimation came with the introduction of OpenPose [2], an open-source framework capable of real-time multi-person 2D pose detection using part affinity fields. OpenPose enables reliable extraction of body keypoints from standard camera footage without depth sensors or markers, making it a foundational tool for gesture-based applications. An alternative to OpenPose is Google's MediaPipe [6], which also provides real-time pose estimation. While both are highly capable, OpenPose was chosen for this work due to its widespread use in academic research and its BODY-25 model, which offers a rich set of keypoints suitable for detailed pose analysis.

In a related domain, He et al. [4] explored the recognition of traffic police gestures using a combination of convolutional pose machines and handcrafted spatial features. Their method also incorporated LSTM networks to model temporal patterns. While their focus was on dynamic gesture sequences, their integration of pose-based features laid important groundwork for gesture interpretation in safety-critical contexts.

Several other studies have leveraged keypoint extraction and deep learning for gesture recognition. Liu et al. [5] employed a Spatio-Temporal Graph Convolutional Network (ST-GCN) with attention mechanisms to achieve high accuracy on a large dataset of police gestures. Similarly, Ma et al. [7] developed a real-time ST-CNN using Kinect data, demonstrating strong performance in virtual city environments. Sathya et al. [10] used cumulative frame differences and a Random Forest classifier for static gesture recognition. Mishra et al. [9] focused on recognizing authorized traffic controllers by first detecting them with an object detector, then reconstructing 3D hand models for gesture classification with a CNN. A related approach for dynamic gesture recognition was proposed by Bagladi et al. [1].

While our work focuses on 2D pose estimation for simplicity and efficiency, 3D pose estimation offers a more direct solution to viewpoint variations. For instance, Fóthi et al. [3] proposed a method for multi-view, multi-body 3D pose estimation that does not require camera calibration. Such methods can inherently handle rotation but often demand more complex models and multiple camera setups, which contrasts with our goal of a lightweight, single-camera system.

Building upon these works, our method targets the recognition of static arm signals using only pose keypoints, without temporal modeling, and emphasizes

robustness against viewpoint variations.

4. The proposed pipeline

The proposed system recognizes static human arm signals through a multi-stage pipeline, as depicted in Figure 1. The process consists of the following main stages:

1. We take 2D pictures of the person giving the static human arm signal.
2. Performing 2D keypoint estimation using the OpenPose BODY-25 model.
3. Normalization of the OpenPose keypoints for scale and position invariance.
4. Classification using a gesture recognition model (see Section 5).

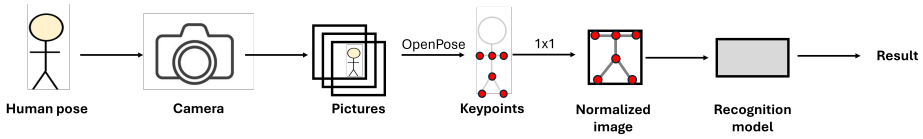


Figure 1. Pipeline for static arm gesture recognition.

The initial stages of the pipeline – image capture, keypoint estimation, and normalization – are detailed in the following subsections.

4.1. Image capture

The process starts with capturing a 2D image of the person performing the gesture using a simple webcamera. This approach requires no additional depth sensors or specialized hardware. To ensure reliable keypoint detection, the person's upper body must be fully visible in the frame.

4.2. Keypoint estimation

The captured image is processed using the OpenPose framework [2] to extract 2D keypoints. We utilize the BODY-25 model to obtain the coordinates of 25 anatomical points, which provides a structured representation of the person's pose for the subsequent steps.

4.3. Normalization

The raw keypoint coordinates from OpenPose are not directly suitable for gesture classification because they are sensitive to the person's position, distance from the camera, and individual body proportions. To address this, we apply a normalization step.

The skeleton is transformed to fit within a unit square, with the neck keypoint (ID 1) serving as the new origin $(0, 0)$. This process of translation and uniform scaling makes the gesture representation independent of the person's size or position in the frame. By removing this variability, the normalization allows the classification models to focus purely on the pose itself.

5. Recognition models

The recognition of static arm gestures in this study is approached using two fundamentally different machine learning paradigms. The first is an unsupervised method based on the K-means clustering algorithm, while the second utilizes a supervised neural network classifier. Both methods operate on normalized 2D keypoint vectors produced by the pose estimation pipeline described in Section 4, and are designed to assign the input pose to one of a finite number of predefined gesture categories.

This subsection focuses on the K-means clustering approach, detailing the configuration phase, distance metrics, evaluation process, and practical considerations of using this method for real-time gesture classification.

5.1. K-means clustering for gesture classification

K-means is an unsupervised learning algorithm widely used for partitioning data into K distinct clusters based on geometric similarity. In the context of gesture recognition, each cluster corresponds to a specific arm pose, and the centroid of that cluster serves as its representative gesture template.

5.1.1. K-means algorithm overview

The standard K-means algorithm proceeds through the following steps:

1. **Initialization:** K cluster centroids are initialized from representative samples.
2. **Assignment:** Each input data point (normalized pose vector) is assigned to the closest centroid using a chosen distance metric.
3. **Update:** New centroids are computed by taking the mean of all vectors assigned to each cluster.
4. **Iteration:** Steps 2 and 3 are repeated until the centroids stabilize.

Once configured, the resulting centroids can be stored and used for efficient classification of unseen samples by identifying the nearest cluster representative, which is called centroid.

Typically, 10-20 samples per gesture were sufficient in my experiments to yield stable and accurate centroids.

5.1.2. Pose representation and distance metrics

Each normalized pose is represented as a 50-dimensional real-valued vector formed by concatenating the (x, y) coordinates of 25 BODY-25 keypoints. Classification is performed by computing the distance between this input vector and each of the K centroids. The sample is assigned to the class of the closest centroid.

We explored multiple distance metrics:

- **Euclidean distance**, calculated over all keypoints,
- **Weighted distance**, where each keypoint contributes with a custom weight to emphasize informative keypoints.

The weighted distance is defined as:

$$\rho(\mathbf{A}, \mathbf{B}) = \sqrt{\sum_{i=1}^{50} w_i \cdot (A_i - B_i)^2}$$

where $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{50}$ are the input vectors, and $\mathbf{w} \in \mathbb{R}^{50}$ is a manually constructed weight vector. Higher weights are typically assigned to the elbows and wrists, which are key to distinguishing gestures. The Euclidean distance is a special case of this formula where all weights w_i are equal to 1.

5.2. Neural network-based classification

In addition to the unsupervised K-means clustering method, we also implemented a supervised neural network approach for static arm gesture classification.

5.2.1. Neural network architecture

The classification model is a feedforward neural network composed of fully connected (linear) layers interleaved with ReLU activation functions. The input layer receives the 50-dimensional normalized pose vector, and the output layer produces class scores for the 9 gesture categories. Several configurations were tested; a typical architecture that achieved high accuracy was as follows:

- Input: 50 features (normalized keypoints)
- Hidden layers: [1024, 512, 256] neurons with ReLU activation
- Output layer: 9 neurons (gesture classes) with softmax activation

5.2.2. Training and validation

The labeled dataset was split into training, validation, and test sets in a 60–10–30% ratio. The model was trained using the Adam optimizer with a batch size of 4096. Training was performed for multiple epochs, with early stopping based on validation accuracy. The stopping threshold was set to 99.5% validation accuracy.

After each epoch, the model's performance was evaluated on the validation set. If the model surpassed the accuracy threshold, training was halted to avoid overfitting.

6. Dataset

To evaluate the proposed recognition methods, we recorded a custom dataset of static human arm signals using a standard webcamera. No special hardware was used, ensuring that the system remains cost-effective and widely deployable.

6.1. Data collection procedure

The dataset was recorded using a standard webcam setup. Images were extracted from videos and then processed through the OpenPose framework to extract 2D pose keypoints.

To ensure robustness and variability, multiple individuals were involved in the data collection process. While the majority of the samples were performed by the author, three additional participants were recruited to enrich the dataset. These contributors received brief verbal instructions about the arm signals but were not trained in any standardized way, resulting in natural variation in gesture execution. This diversity helps the models generalize across different body types and styles of gesture performance.

6.2. Recorded signals

We defined a total of nine distinct static arm signals, inspired by standardized traffic control and hand signaling conventions. These are:

Each sample in the dataset was manually labeled according to one of the categories above. Participants held the same arm position with minor natural variation for several seconds, from which frames were extracted to increase the sample count.

A total of 53000 labeled samples were collected for the full dataset.

6.3. Rotated dataset

In practical applications, it is common for the individual giving a signal not to be directly facing the camera (Figure 3), which may result in reduced accuracy. This paper presents various approaches to address the problem of rotated signals. One solution involves artificially generating rotated data by estimating the depth of the keypoints from the data captured facing the camera, producing a 3D skeleton that can be rotated in 3D space before being projected back onto a 2D plane. Since only the data captured facing the camera is required for this process, there is no need to create additional datasets, making the solution both convenient and easy to use. For the K-means algorithm, the centroids are augmented with this synthetic data after the initial configuration phase. Similarly, this artificially generated training data can also be employed in training the neural network.

To test the robustness of the system against moderate body rotation, we also recorded a second, smaller dataset focusing on rotated poses. For this purpose, a custom-printed circular rotation guide (Figure 4) was placed on the floor to allow consistent measurement of the body's rotation angle relative to the camera.

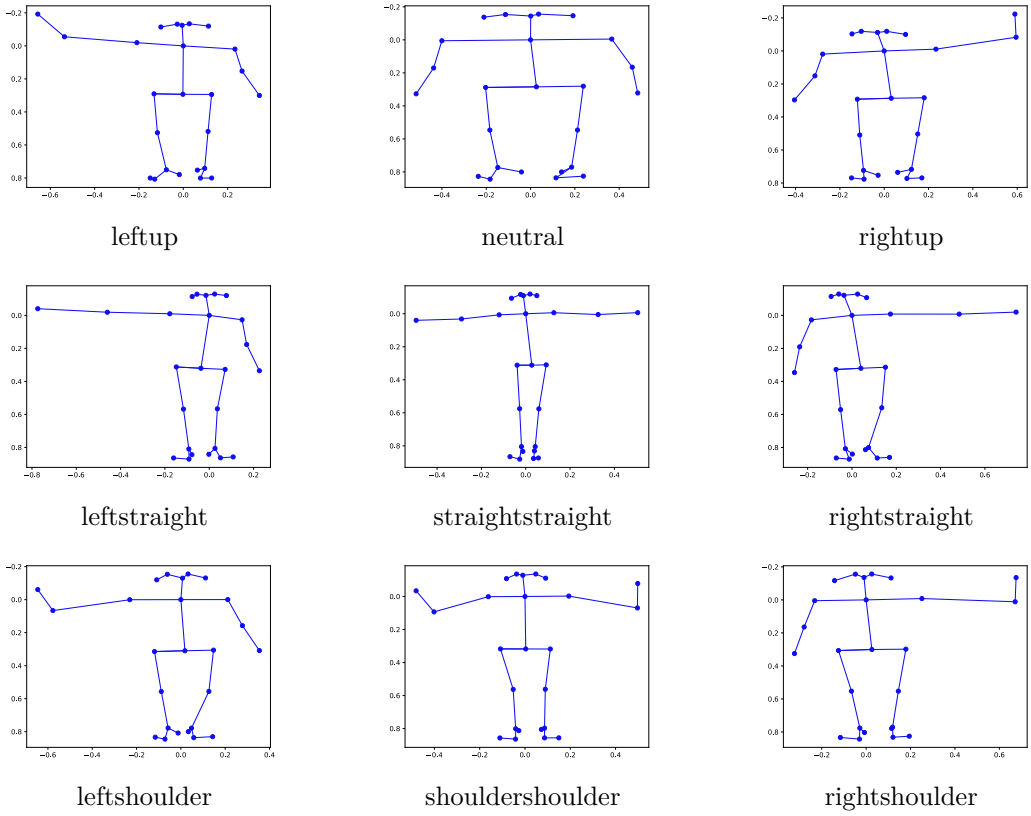
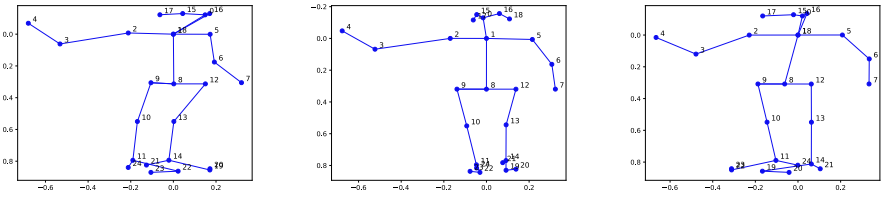


Figure 2. The nine static arm signals in the dataset.



(a) 45° to the left **(b)** Facing the camera **(c)** 45° to the right

Figure 3. Examples of rotated static arm signals.

Each subject stood on the guide and performed the signal while rotated by specific angles. This setup ensured that the rotated dataset was captured with high angular precision.

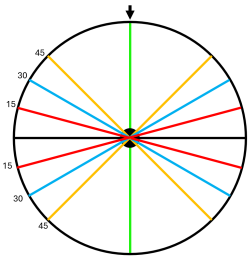


Figure 4. Custom-made rotation guide for precise data collection.

We selected a representative subset of three arm signals for rotation: `neutral`, `rightshoulder`, and `rightup`. These were recorded at rotation angles of 15°, 30°, and 45° both to the left and right relative to the frontal view. This setup enabled controlled testing of the recognition system using real-world data. The rotated dataset consists of over 17,000 front-facing samples and over 34,000 rotated samples. Our results show that these solutions robustly handle rotated hand signals up to 45° with respect to the ideal case of facing the camera, while maintaining benefits like speed and accuracy in recognition (Figure 7).

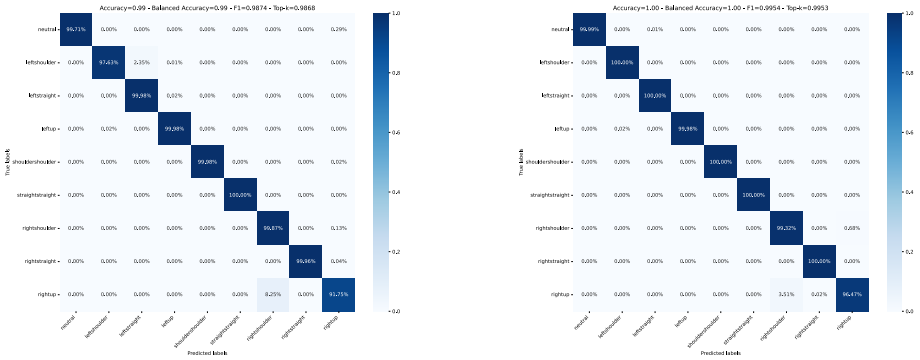
7. Results

The performance of the proposed gesture recognition methods was evaluated on both the full dataset of nine static arm signals and the rotated subset of three gestures. The results are summarized in this section.

7.1. Full dataset results

The K-means clustering approach demonstrated high accuracy in classifying the nine static arm signals. As shown by the confusion matrices in Figure 5, both the standard Euclidean distance and the weighted distance metric yielded excellent results with minimal confusion between gestures. The weighted distance, which emphasizes keypoints on the arms and hands, provided a marginal improvement, confirming the effectiveness of this simple, unsupervised method for pose classification.

Similarly, the neural network classifier achieved outstanding performance on the full dataset. The confusion matrix in Figure 6 illustrates that the model correctly classified nearly all test samples, demonstrating its capacity to learn robust representations of the gestures. Both methods proved to be accurate and effective for recognizing the defined set of static arm signals.



(a) K-means clustering results on the full dataset (Euclidean distance).

(b) K-means clustering results on the full dataset (weighted distance).

Figure 5. Confusion matrices for K-means clustering on the full dataset.

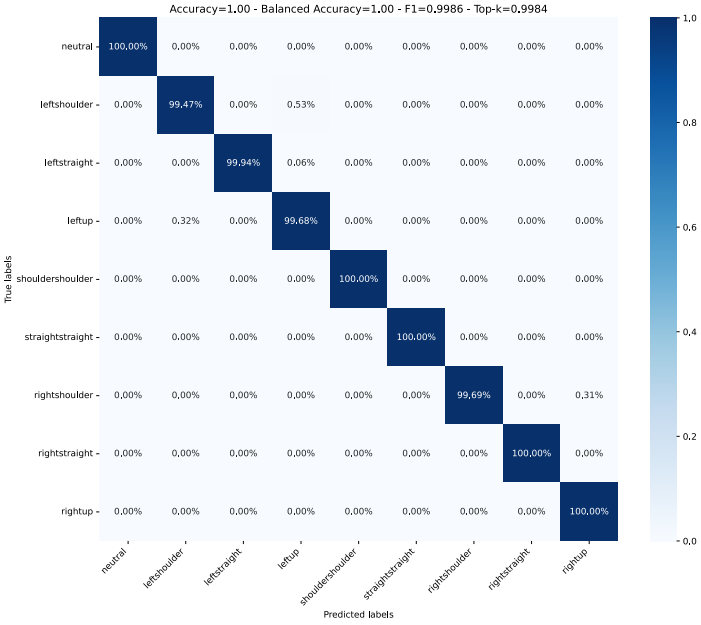
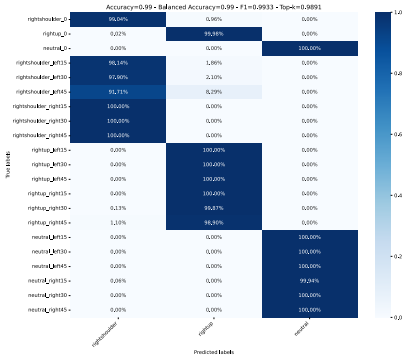


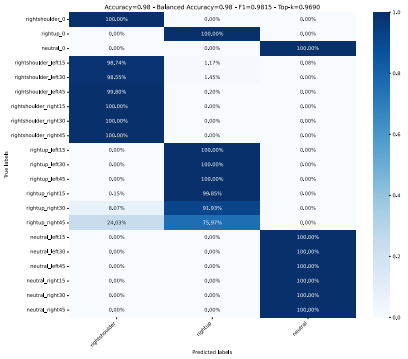
Figure 6. Neural network classifier results on the full dataset.

7.2. Rotated dataset results

The robustness of both the K-means and neural network models against body rotation was evaluated using the rotated dataset. The results, summarized in Figure 7, demonstrate that both methods maintain high recognition accuracy even when the subject is rotated up to 45° from the frontal view. While minor misclassifications occur at larger rotation angles, particularly for the K-means model, the overall performance is highly successful. This confirms that our approach, including the use of artificially generated rotated data for training, effectively solves the challenge of viewpoint variation in static gesture recognition.



(a) K-means clustering results on rotated dataset.



(b) Neural network results on rotated dataset.

Figure 7. Confusion matrices of K-means and neural network on the rotated dataset.

8. Conclusions

This paper presented a method for recognizing static human arm signals using 2D keypoint estimation and machine learning classification. The approach is based on OpenPose for keypoint extraction, followed by normalization and classification using K-means clustering or a neural network. A novel aspect of the work is the generation of artificially rotated data to augment the training set, improving the model's robustness to changes in body orientation. The methods were evaluated on a custom dataset of nine gestures, with results showing high accuracy and efficiency even with body rotations of up to 45 degrees, indicating the potential for practical applications in real-time gesture recognition using standard camera equipment.

Acknowledgements. Supported by the ÚNKP-23-6 New National Excellence Program of the Ministry for Culture and Innovation from the source of the National Research, Development and Innovation Fund.

References

[1] M. Z. Bagladi, L. Gulyás, G. Szalay: *Fast Real-Time Pipeline for Robust Arm Gesture Recognition*, in: Proceedings of the Intelligent Robotics FAIR 2025, IntRob '25, Association for Computing Machinery, 2025, pp. 138–143, ISBN: 9798400715891, DOI: 10.1145/3759355.3759633.

[2] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, Y. Sheikh: *OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields*, 2019, arXiv: 1812.08008 [cs.CV].

[3] Á. Fóthi, J. Skaf, F. Lu, K. Fenech: *Deep NRSFM for multi-view multi-body pose estimation*, Pattern Recognition Letters 185 (2024), pp. 218–224, ISSN: 0167-8655, DOI: 10.1016/j.patrec.2024.08.015, URL: https://www.sciencedirect.com/science/article/pii/S0167865524002472.

[4] J. He, C. Zhang, X. He, R. Dong: *Visual Recognition of traffic police gestures with convolutional pose machine and handcrafted features*, Neurocomputing 390 (2020), pp. 248–259, ISSN: 0925-2312, DOI: 10.1016/j.neucom.2019.07.103, URL: https://www.sciencedirect.com/science/article/pii/S0925231219314420.

[5] K. Liu, Y. Zheng, J. Yang, H. Bao, H. Zeng: *Chinese Traffic Police Gesture Recognition Based on Graph Convolutional Network in Natural Scene*, Applied Sciences 11.24 (2021), ISSN: 2076-3417, DOI: 10.3390/app112411951, URL: https://www.mdpi.com/2076-3417/11/24/11951.

[6] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg, M. Grundmann: *MediaPipe: A Framework for Building Perception Pipelines*, 2019, arXiv: 1906.08172 [cs.DC], URL: https://arxiv.org/abs/1906.08172.

[7] C. Ma, Y. Zhang, A. Wang, Y. Wang, G. Chen: *Traffic Command Gesture Recognition for Virtual Urban Scenes Based on a Spatiotemporal Convolution Neural Network*, ISPRS International Journal of Geo-Information 7.1 (2018), ISSN: 2220-9964, DOI: 10.3390/ijgi7010037, URL: https://www.mdpi.com/2220-9964/7/1/37.

[8] E. Mathe, A. Mitsou, E. Spyrou, P. Mylonas: *Arm Gesture Recognition using a Convolutional Neural Network*, in: 2018 13th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP), 2018, pp. 37–42, DOI: 10.1109/SMAP.2018.8501886.

[9] A. Mishra, J. Kim, J. Cha, D. Kim, S. Kim: *Authorized Traffic Controller Hand Gesture Recognition for Situation-Aware Autonomous Driving*, Sensors 21.23 (2021), ISSN: 1424-8220, DOI: 10.3390/s21237914, URL: https://www.mdpi.com/1424-8220/21/23/7914.

[10] R. Sathya, M. K. Geetha: *Framework for Traffic Personnel Gesture Recognition*, Procedia Computer Science 46 (2015), Proceedings of the International Conference on Information and Communication Technologies, ICICT 2014, 3-5 December 2014 at Bolgatty Palace Island Resort, Kochi, India, pp. 1700–1707, ISSN: 1877-0509, DOI: 10.1016/j.procs.2015.02.113, URL: https://www.sciencedirect.com/science/article/pii/S1877050915001775.